S O F T S Y N T H™

digidesign

SOFTSYNTH™

USER'S MANUAL

digidesign
1360 Willow Road
Suite 101
Menlo Park, CA 94025
Tel: (415) 327-8811

CONTENTS

4 ABOUT THIS MANUAL

5 INTRODUCTION
5 Registration
5 Telephone Support
6 Updates

7 SECTION 1 - GETTING STARTED
7 What You'll Need
7 Hard Disks
7 Using Softsynth with the Emulator II
7 MIDI Interface
7 Copy Protection
8 Key Disk
8 Double-sided Drives
8 Single-sided Drives
9 Connecting the MIDI Interface
9 Serial Port/Clock Speed
10 Selecting a Sampler
10 Softsynth File Types
11 Sound Designer™ Compatibility

13 SECTION 2 - ABOUT ADDITIVE SYNTHESIS
13 What is Additive Synthesis?
14 Fundamental Frequency
14 Partials
14 Harmonic Series
15 Ratio
16 Sine Waves
17 Envelope
17 Complex and Simple Envelopes
18 Tuning Contour

19 SECTION 3 - USING SOFTSYNTH
19 Starting Softsynth
19 The Main Screen
20 Viewing Angle
21 Frequency
21 Sample Rate
22 Sound Length
22 The Harmonic Faders
23 Single Partial Editing

23 The Envelope Window
24 Breakpoints
26 Tuning Contour
27 Ratio
27 Harmonic Series
28 Wave Types
29 Editing Techniques
30 Editing Tools (*Copy, Paste, Clear*)
31 Preview (Mac speaker playback)
31 Single Partial Preview
32 Synthesize
32 Normalize
33 Transferring Sound Files to the Sampler
33 MIDI Keyboard
34 Time Slice Editing
34 Timbre Events
35 Master Envelope
35 Timbre Event Bar
36 Creating Timbre Events
37 Moving Timbre Events

39 SECTION 4 - THE SMARTSYNTH FUNCTION
39 How Does it Work?
40 Creating a New Sound
40 Parameter Types
41 Harmonic Series
42 Harmonic Range
42 Harmonic Filter
43 Partial Detuning
43 Doubling
44 Frequency Movement
45 Attack Rate
45 Frequency Attack
46 Percussion
46 Decay Rate
47 Sustain Level
47 Fade
48 Using Smartsynth to Modify an Existing Sound

49 SECTION 5 - SAMPLER PERSONALITIES
49 About Samplers
51 Sequential Prophet 2000/2002
53 Ensoniq Mirage/Multisampler
55 E-mu Systems Emulator II
57 Korg DSS-1
59 Akai S900
61 Akai S612

63 SECTION 6 - TIPS AND TRICKS
63 Start with a Parameter File Copy
63 Faster Synthesis
64 Creating Presets
64 Velocity Effects
65 Sound Designer Tips
65 Crossfade Looping Detuned Sounds
65 Using Frequency Analysis
66 Combining Softsynth and Karplus -Strong

67 SECTION 7 - CASE STUDIES
67 Programming Style
67 Send us your Case Studies!

A1 Appendix A - Hard Disk Installation

A2 Appendix B - Note to Frequency Conversion Chart

A3 Appendix C - Softsynth Credits

A4 Appendix D - Additional Sampler Personalites

Supplement: FM Synthesis

About This Manual

This manual assumes that you have a basic understanding of both the Macintosh and your digital sampler. If you are not familiar with the Macintosh, please take some time to learn standard Macintosh terms and techniques before using Softsynth.

The **Introduction** describes Digidesign's update and telephone support policies, and urges you to send in your registration card.

Section 1, Getting Started, covers basic hardware requirements and proper system configuration. Softsynth file types and Sound Designer file compatibility are also discussed.

Section 2, About Additive Synthesis, is a basic tutorial on the principles of additive synthesis. Read this section - it defines terms that are used throughout this manual.

Section 3, Using Softsynth, teachs you to create sounds using Softsynth's *Single Partial* and *Time Slice* editing modes.

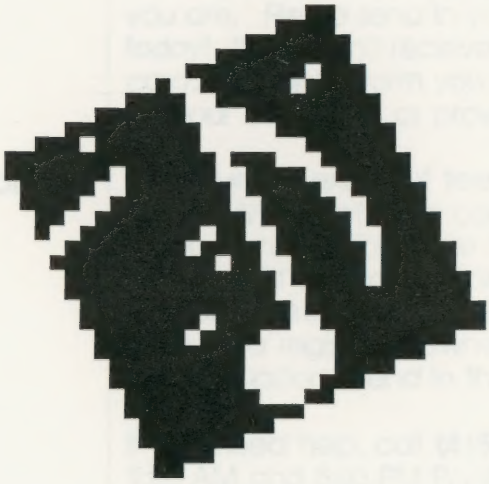
Section 4, The Smartsynth Function, explains Softsynth's unique Smartsynth function, which generates new sounds based on general sound characteristics you select.

Section 5, Sampler Personalities, describes each Softsynth compatible sampler, and explains how to transfer Softsynth sounds to your sampler. See App. D for information on additional Softsynth compatible samplers.

Section 6, Tips and Tricks, contains various user tips, shortcuts, etc.

Section 7, Case Studies, includes several user's descriptions of the techniques they used to create a Softsynth sound.

Introduction



Hello!

Congratulations on your purchase of Softsynth software! Softsynth is a new, unique type of sound creating tool - you will be one of the first to create sounds in software on the Macintosh.

Our goal at Digidesign is to create interesting, useful "tools" for musicians. You can help by giving us feedback on our programs - do they work for you? How can they be improved? What features would you like to see? What new types of programs would you like us to develop? We prefer written suggestions - it helps to have something written, rather than relying on our memories!

Registration

It is very important that you let us know who you are. *Please* send in your registration **today!** If we don't receive your registration card, we can't inform you of updates, send you our newsletter, or provide support.

Telephone Support

We have an excellent telephone support policy. Many software companies now charge an additional fee for telephone support, or limit the amount of free support. Our support is free and unlimited. However, you **must** be a *registered* owner to qualify for telephone support - send in that card today!

If you need help, call **(415) 327-8811** between 9:00 AM and 5:00 PM Pacific Standard Time. Please read this manual carefully before calling for help.

You will need your serial number when you call for telephone support - it is written on the back side of your master program disk.

Updates

Future Softsynth program updates will be available to registered owners for a nominal price. However, to receive information about program updates, you must be a registered owner. Have you sent that card in yet?

If you plan to use Softsynth with a hard disk, read Appendix A, "Hard Disk Installation" care-

Section 1

Getting Started

What You'll Need...

In addition to your Softsynth software, you will need a 512k Macintosh or Macintosh Plus, a Macintosh MIDI interface and a Softsynth-compatible digital sampling instrument. See the back panel of the Softsynth package for a list of digital sampling instruments that are compatible with this version of Softsynth.

If you have a 512K Mac with a single-sided internal disk drive, you will also need an external drive or a hard disk. Macintosh Plus and Macintosh 512K Enhanced owners will not need an external drive or hard disk.

Hard Disks

In general, a hard disk has many advantages for running Softsynth. Digital sound files are very large, and they will quickly fill a floppy disk. Hard disks also run faster than floppy disks, reducing the amount of time required to generate a set of new sounds.

Using Softsynth with the Emulator II

Emulator II owners must have Sound Designer to use Softsynth. If you plan to use Softsynth with an Emulator II, read section 5 "Sampler Personalities" now.

MIDI Interface

Softsynth will work with all commercially available Macintosh MIDI interfaces. However, if you are using a Macintosh Plus, you must have a Macintosh Plus compatible MIDI interface.

Softsynth is supplied with one master program disk and one sound parameters disk. Make a back-up copy of each disk.

Copy Protection

The master program disk is copy protected using "key disk" protection. Each time you start up the Softsynth program you must insert the master disk. You will not be required to insert the master disk again until you quit and re-start the program.

The Softsynth program can be installed on a single hard disk , bypassing the key disk copy protection (the program can be run without inserting the master program disk). See Appendix A, "Hard Disk Installation" for instructions.

Working Copy

If you do not have a hard disk, you should run Softsynth from a *copy* of the master program disk. **Do not** run Softsynth from the master disk - keep it "locked" and use it only as a "key" disk!

Key Disk

When you start Softsynth from a copy of the master disk, the program will ask you to insert your master disk (the "key disk"). A few seconds after you have inserted the key disk it will be automatically ejected and the program will start.

Double-sided Drives

If you have a double-sided internal disk drive (Macintosh Plus or Macintosh 512K enhanced), copy the Softsynth master disk to a disk that has been formatted as "two-sided" (800K capacity).

Single-sided Drives

If you have a single-sided internal disk drive (Macintosh 512K), copy the Softsynth master disk to a disk that has been formatted as "one-sided" (400K capacity). You will need an external drive connected with a blank, formatted disk inserted for storing Softsynth's *sound* and *parameter* files.

Connecting the MIDI Interface

Plug the MIDI interface into either of the Macintosh's serial ports (**printer** or **modem**). Use whichever port is convenient - Softsynth can use either port.

Connect the MIDI Out jack of the Macintosh MIDI interface to the MIDI In jack on the sampler, and the MIDI In jack of the Macintosh MIDI interface to the MIDI Out jack on the sampler.

The hardware is now interfaced and ready to go. Refer to Section 5, "Sampler Personalities" to determine if your sampler requires any additional set up.

Turn on the Macintosh first, then the Sampler. Use your copy of the Softsynth program disk as your startup disk. Double click on the Softsynth program icon. After you have inserted your key disk and re-inserted your copy, Softsynth's main programming screen will appear.

Setting the Serial Port and Clock Speed

Our first step is setting Softsynth's MIDI parameters. Select **MIDI Setup** from the pull-down **Sampler** menu. A dialog box will appear for selecting the **Serial Port**, and setting the **Clock Speed** of your MIDI interface.

Click on the button next to the icon that represents the serial port where your MIDI interface is plugged in. Click on the button next to the clock speed your MIDI interface requires. Most Macintosh MIDI interfaces use 1mHz clock speeds. Early Southworth interfaces and most Musicworks interfaces use 500 kHz clock speeds. Contact the manufacturer if you are not sure of your interface's clock speed.

After you set the Serial Port and Clock Speed, click on the **OK** button to return to the main screen.

Selecting a Sampler

Next, select the sampler you will use with Softsynth from the list in the pull-down **Sampler** menu. A checkmark will appear next to the sampler's name when it is selected.

The MIDI Setup and Sampler settings are stored when you quit the Softsynth program you will not have to set them again unless you use Softsynth with a different sampler or MIDI interface.

Softsynth File Types

Before you use Softsynth, it is important to understand the two basic types of files that Softsynth creates: **Parameter files** and **Sound files**.

Synthtrumpet

Parameter files store all the parameter settings (envelopes, tuning, etc.) that Softsynth uses to create a digital sound file that can be transferred to a sampler for playback. You create these parameters using Softsynth's programming screens, then tell Softsynth to create a sound file from them.

After a sound file is created (synthesized), changes can be made to the parameter file, and a new sound file can be synthesized. When a parameter file is closed, you have the option to save changes made to it. The parameter file can be re-opened later to change (and then resynthesize) the sound. Parameter files are always the same size (about 30K bytes)

Vibe G2

Sound files consist of digital samples that have been generated by Softsynth. Sound Files can not be "opened" into the Softsynth program - they can only be transferred to the sampler under Softsynth's control.

Sound Designer™ Compatibility

Softsynth sound files are stored in the standard *Sound Designer* 16 bit format. If you have a *Sound Designer* program, you can directly open sound files created using Softsynth.

When a sound file is transferred to the sampler by Softsynth, it is automatically converted to the format (i.e. 8 bit, 12 bit, etc.) required by the selected sampler.

Section 2

About Additive Synthesis

What is Additive Synthesis?

The digital synthesis technique Softsynth uses to create sounds is *Additive Synthesis.* Additive synthesis is generally considered to be the most flexible and powerful digital synthesis technique, but the hardware required to build an additive synthesizer is very expensive. As a result, few hardware based additive synthesizers exist.

Softsynth is a software simulation of a digital additive synthesizer. To understand how Softsynth works, we must understand the basic principles of sound and additive synthesis.

Sound consists of vibrations transmitted through the air. These vibrations are *created* by the *movements* of a sound source such as a voice or a musical instrument. These movements (i.e. the vibration of a vocal chord or a guitar string) create tiny pressure waves in the air. When these pressure waves reach our ears, our eardrums vibrate and we perceive sound.

If we convert these pressure waves to electrical impulses using a microphone, the pressure wave become an electrical voltage. Changes in the intensity and direction of the pressure waves become changes in the voltage level and polarity (positive or negative). The result is an electronic waveform that represents the sound, and can be converted back into air pressure waves using a speaker.

Fundamental Frequency

Sound waveforms that follow a repeating pattern have a characteristic known as **pitch**, and are called *periodic* waveforms. The pitch of a sound is determined by its **fundamental frequency** - the number of repeated vibrations per second. Frequency is expressed in Hertz (Hz), or cycles per second. For example, middle C on a piano produces a sound wave with a fundamental frequency of 264 Hz. The healthy, young human ear can hear frequencies that range from approximately 20 Hz to 20,000 Hz.

Partials

Virtually all sounds contain more than one frequency. They have a fundamental frequency that is the apparent "pitch" of the sound, and one or more (usually many more) **partials** (or **harmonics**). The waveforms of these sounds are called **complex waveforms**.

The partials are additional frequencies that occur above (or sometimes below) the fundamental frequency. Partials are not heard as separate pitches - they determine the timbre, or tonal quality, of the sound.

Harmonic Series

The partials of pitched sounds have frequencies that are based on multiples of the fundamental frequency. Typically, a sound with a fundamental frequency of 440 Hz will have a partial at 880 Hz (2 X 440), 1320 Hz (3X440), etc. The fundamental (440 Hz) is called the first harmonic, 880Hz is the second harmonic - the pattern continues for all partials in the sound's **harmonic series**.

Generally speaking, if the partial is "in tune" (a multiple of the fundamental frequency), it is called a harmonic.

The pitch of each partial is not always an exact whole number multiple of the fundamental. Some sounds have partials that are **enharmonic** - a bell, for example, might have a second harmonic that is 1.5 times the fundamental frequency. This partial is enharmonic, because it is not an exact (or near) whole number multiple of the fundamental.

Many sounds have partials that are slightly "out of tune" with the fundamental frequency. A piano has some partials that are slightly sharp, and some that are slightly flat. These slight variations from the standard harmonic series play an important role in the piano's unique sound.

Ratio

The exact frequency of a partial is often described by its *relationship* to the fundamental frequency. The **Ratio** of a partial is the number you multiply times the fundamental to get the partial's exact frequency. For example, in the standard harmonic series, the second partial has a ratio of 2, the third partial has a ratio of 3, etc. If the second partial has a ratio of 1.95, it is slightly flat (lower in pitch). A ratio of 2.05 would indicate a slightly sharp second partial.

Partials that are less than 2% "out of tune" are generally consider to be harmonic. Partials that are more than 2% out of tune (enharmonic) begin to sound more dissonant, "clangy" and metallic.

Sine Waves

The simplest type of waveform is called a sine wave. A sine wave is a perfectly "pure" tone that does not have any harmonics besides the fundamental. The sine wave is the key to additive synthesis. **All complex waveforms can be broken down into a series of sine waves at different frequencies and amplitudes (loudness levels).**

Sound is similar to light. If we pass sunlight (complex light) through a prism, it is divided into the colors of the spectrum. We cannot see these individual colors when they are part of the sunlight - they combine to create an entirely different type of light. If we combine all of the colors of the spectrum together, the result is "complex" white light.

Likewise, complex sounds can be broken down into their component sine waves, and can be "built" by combining sine waves of the correct frequencies and amplitudes. Softsynth is a "sound building" tool.

The timbre (tone quality) of a sound is determined by the frequencies of the partials (the *harmonic series*) and the relative amplitude of each partial (the *harmonic mix*).

Additive synthesis mixes sine waves and controls changes in their amplitudes as time passes. If we can control the frequency and amplitude of enough sine waves, we can recreate virtually any sound, and create some interesting new sounds as well.

Softsynth gives you precise control over both the initial *ratio* of each partial, and changes in the frequency and amplitude of each partial as time passes. These are the building blocks of additive synthesis.

Envelope

Changes in the amplitude of a partial as time passes is called the **envelope** of the partial. Some simple additive synthesis techniques use a single envelope to control the overall amplitude of the entire complex sound. This approach is not very useful because the envelopes of each partial change quite differently from each other in most sounds.

Softsynth provides a separate, independent envelope for each partial. This envelope can be adjusted with an accuracy of approximately 2 milliseconds (thousandths of a second).

Complex and Simple Envelopes

Envelopes can also be simple or complex. A typical analog synthesizer has a simple envelope generator with four stages: attack, decay, sustain and release. Softsynth provides a complex 40 (!) stage envelope for each partial. These complex envelopes allow you to create very elaborate sounds that are not possible with most analog or digital synthesizers.

The most complex part of most sounds is the attack. A piano, for example, has many partials at high amplitudes during the first 50 milliseconds. Within about 100 milliseconds, many of the higher partials have completely faded out. The attack characteristics play an important role in the overall effect of a sound.

Tuning Contour

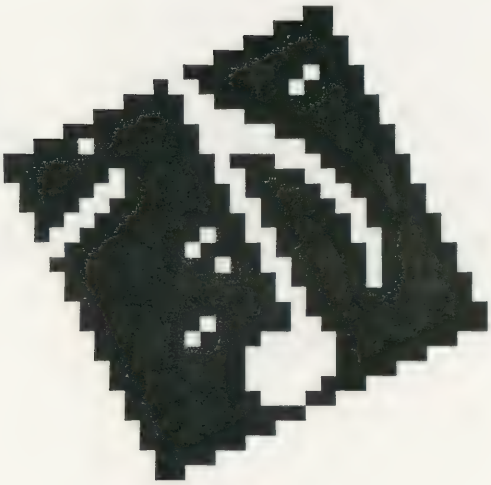
In addition to a complex envelope, each Softsynth partial also has a **tuning contour**. The tuning contour controls changes in the actual pitch of the partial as time passes.

In many natural sounds, the pitch of both the fundamental and individual partials will fluctuate. Many sounds have partials that go sharp and/or flat during the attack only. For example, when a trumpet is blown, many partials go sharp, then flat, then "in tune" within only the first few milliseconds of the sound. These fluctuations are responsible for the "bwah" sound at the start of a loud trumpet note.

Softsynth's partials can deviate from their basic pitch by up to +/- 50%. Minor variations of +/- 1% will sound best with most sounds, creating a "chorusing" or detuning effect. Large amounts of pitch fluctuation will cause audible "pitch sweeps" in the sound.

Section 3

Using Softsynth

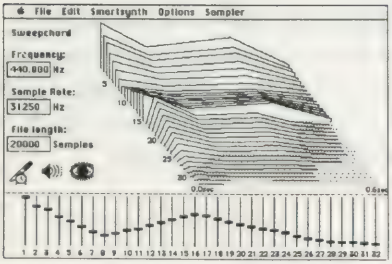


Starting Softsynth

Now that we have reviewed the basic concepts of additive synthesis, we're ready to start creating sounds using Softsynth. If you have not already set up your Softsynth system, please do so now (see Section 1, "Getting Started").

Once your Softsynth System is connected, start the Softsynth program by double clicking the mouse on the Softsynth program icon. Use a *copy* of the master program disk - the Macintosh will ask you to insert your original master disk (the "key" disk).

The Main Screen

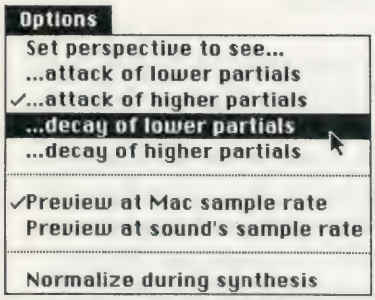


In a few seconds, Softsynth's **Main Screen** will appear on the Macintosh.

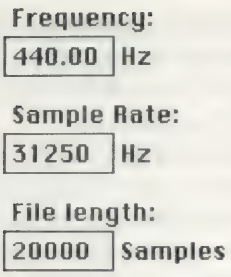
The Main Screen displays the envelopes of all 32 partials, and the harmonic faders that are used to adjust the overall amplitude of each partial. However, we have not created any envelopes or set the levels of the partials, so nothing (except rows of dots) will appear in the main screen's partial envelopes display. To see a typical display of partial envelopes, close the "blank" parameter file shown on the screen (select **Close** from the **File** menu), and open one of the parameter files supplied on the Softsynth parameter files disk.

The main screen displays all of the partial envelopes (from 1 to 32) "stacked" from the front to the back of the Macintosh screen. The numbers of the partials (in increments of 5) are listed to the side of the display. The envelopes are drawn in the standard style - time runs from left to right, and height represents changes in the amplitude (or loudness) of the partial. The overall height of each partial is determined by its envelope and the level of its harmonic fader (located below the partial envelopes display).

Viewing Angle



The partial envelopes display can be viewed from four different angles. This feature is useful if low amplitude partials are not visible because they are "behind" higher amplitude partials. A new viewing angle can be selected from the **Options** menu. The viewing angles are listed by the envelope areas they display best, (for example "Show attack of higher partials"). A check mark appears next to the currently selected viewing angle.



Three rectangular parameter boxes labeled **Frequency**, **Sample Rate** and **Sound Length** appear to the left of the partial envelopes display. New values can be entered in each of these boxes using standard Macwrite style editing (i.e. select the value and backspace to delete it, then type in the new value).

Frequency

The **Frequency** parameter is the fundamental frequency (the frequency of the first partial) in Hertz (Hz) of the Softsynth sound. The frequencies of all other partials are based on this frequency. Enter a frequency that corresponds to the musical note you want to synthesize. "A" is a particularly easy note to synthesize because its frequency is a multiple of 440 hertz (i.e. 220 Hz, 880 Hz, 110 Hz, etc.). 440 Hz is the default setting for the frequency parameter.

Sample Rate

The **Sample Rate** parameter determines the number of samples in each second of the Softsynth sound. A default value is automatically entered when you select a sampler from the Sampler menu. This value is the most commonly used sample rate for the selected sampler. However, many samplers have variable sample rates. It is important to match the *playback sample rate* of your sampler to the Softsynth Sample Rate parameter to maintain the proper pitch. See Section 5, "Sampler Personalities" for a description of each Sampler's playback sample rates.

Sound Length

The **Sound Length** parameter determines the overall length of the Softsynth sound (in *number of samples*). The actual length (in *seconds*) of the Softsynth sound is determined by both the sound length and the sample rate. For example, if the Sample Rate is 30,000 Hz (samples per second), then a 30,000 sample sound would be 1 second long, a 60,000 sample sound would be 2 seconds long, etc. If we increase the sample rate and keep the number of samples the same, the sound will become shorter.

The Harmonic Faders

A row of 32 Harmonic Faders appears along the bottom of the main screen. These faders adjust the *overall* amplitude of each individual partial.

A small number appears below each fader indicating the partial it controls (i.e. fader 3 controls partial 3). To adjust the amplitude of a partial, click on its fader's horizontal "handle", hold the mouse button down, and "drag" the handle up or down. These faders work like the faders on an audio mixing board - *raising* the fader *increases* the partial's amplitude, and *lowering* the fader *decreases* the partial's amplitude. As you adjust a fader, its level (from 0 to 63) is displayed to the right of the harmonic faders.

Click once on the **Eye** icon to update the partial envelopes display after you have adjusted one or more harmonic faders. The display will update automatically if you switch to another programming screen (more on this later), then return to the main screen. However, if you remain in the main screen, you must click on the eye icon to "see" an updated partial envelopes display.

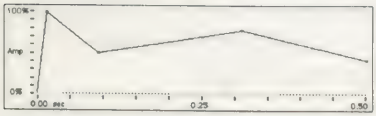
Single Partial Editing

Single Partial editing mode is used to edit the *amplitude envelope, tuning contour, wave type* and *ratio* of each partial.

To enter Single Partial mode, click the cursor once on the fader number of the partial you wish to edit. For example, if you want to edit partial 3, click on the small "3" below the third fader.

The Single Partial programming screen will appear for the individual partial you selected. You can return to the main programming screen at any time by clicking once on the **Mixer** icon.

The Envelope Window



The top half of the screen displays a graph of the selected partial's envelope (changes in its amplitude as time passes). This is the **Envelope Window.**

Amplitude levels (from 0 to 100% of full scale) are displayed along the vertical axis on the left side of the graph.

Time is indicated below the envelope graph. A segment of the envelope approximately one-half second long is displayed - use the scroll bar below the envelope graph to display other sections of the envelope.

The envelope is represented by a series of line segments collectively called the **Envelope Line**. This line traces changes in the partial's amplitude as time passes.

If an envelope has not been created for the partial, the envelope line will appear as a straight line along the bottom of the envelope window.

Breakpoints

The envelope line consists of a group of straight line segments joined by **Breakpoints**. Breakpoints appear as tiny black boxes along the envelope line.

Each envelope can have up to 40 breakpoints. These breakpoints can be placed anywhere in the sound, with an accuracy of approximately 4 millseconds.

Breakpoints are easy to edit. To create or change a breakpoint, first click once on the **Edit Breakpoint** icon (a hand pulling on a rubber band).

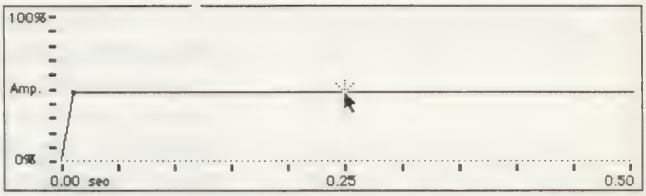
To *create* a *new* breakpoint, position the cursor **on** the envelope line, click the mouse, and drag the envelope line to a new time and/or amplitude. Imagine that the envelope is a rubber band, and you are stretching it to a new position. The new breakpoint will be created when you release the mouse button. A small black box will appear at the actual breakpoint.

To *change* an existing breakpoint, click on the breakpoint's black box and drag it to a new position. You cannot drag a breakpoint before the previous breakpoint (earlier in time), or after the following breakpoint (later in time).

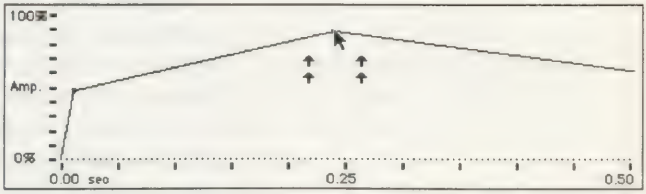
To *delete* a breakpoint, click once on the **Erase Breakpoint** icon (the eraser), then click on the "black box" of the breakpoint you want to delete. You only have to click on the Erase Breakpoint icon before erasing the first breakpoint - the cursor will stay in "erase mode" until you click once on the Edit Breakpoint icon.

How to create a breakpoint:

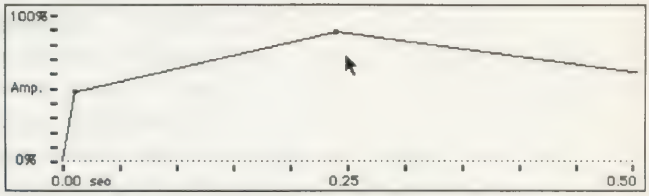
Position the cursor on the envelope line



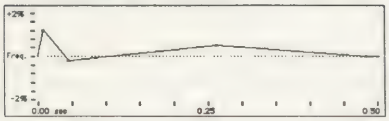
Hold the mouse button down and drag the envelope line to the desired position.



Release the mouse button. The new breakpoint will appear as a small box .



Tuning Contour



The **Tuning Contour** window appears in the bottom half of the Single Partial programming screen, directly below the Envelope window.

The tuning contour graph is similar to the envelope graph - it consists of straight line segments connected at breakpoints. These breakpoints can be created, edited and deleted using the same techniques used to edit the envelope graph.

The tuning contour line displays *changes* in the partial's pitch as time passes.

The **zero line** that runs horizontally across the middle of the tuning contour window represents the partial's basic pitch (the frequency shown in the **Ratio** box at the top of the screen). As the tuning contour line moves *above* the zero line, the pitch becomes *higher* (sharper). As the tuning contour line moves *below* the zero line, the pitch becomes *lower* (flatter).

The **Range** of pitch change provided by the tuning contour window can be either +/- 2% (for fine pitch variations), or +/- 50% (for major pitch changes). The +/- 2% range is useful for imitating the subtle fluctuations of harmonics in natural sounds, or creating chorusing effects. The +/- 50% range is useful for fast/wide pitch changes on the attack of sounds such as trumpets, or for drum sounds with long fall-off.

☞ **Once you have adjusted the pitch range at the 50% setting, you should not switch to the 2% setting!**

Select the pitch change range by clicking on the appropriate button to the left of the tuning contour window.

Ratio

The basic pitch of the selected partial is determined by the **Ratio** parameter. The pitch of each partial is a *multiple of the fundamental frequency*. The Ratio is the number that the fundamental frequency is multiplied by to determine the pitch (in Hertz) of the partial.

Confused? Let's consider some examples. If our fundamental frequency (entered in the main screen - see page 2) is 440 Hz, and the ratio of the third partial is 3.5, then the third partial's frequency (pitch) will be 440 X 3.5, or 1540 Hz.

The default settings for the partials are the same as the partial's numbers. For example, the second partial's ratio is two, the third partial's ratio is three, and the thirty-second partial's ratio is thirty-two.

Harmonic Series

The ratios of all the partials, considered as a group, is called the **Harmonic Series** (see Section 2). The harmonic series helps determine the overall timbre, or tonal quality, of a sound. The default harmonic series contains the typical partial ratios of an organ. This does not necessarily mean that you can only create organ-like sounds with these ratios - the partial envelopes and tuning contours can be used to create more "interesting" sounds. However, to take advantage of Softsynth's power you should experiment with different harmonic series.

To change the ratio of a partial, click on the current ratio number at the top of the screen and edit it using standard Macintosh editing techniques (backspace over the old number and type in a new one).

Wave Types

Traditional additive synthesis uses only sine waves for each partial. To provide more flexibility, Softsynth provides five different **Wave Types** that can be independently used for each partial. For example, partial #1 can be one wave type, partial #2 a different wave type, etc.

Wave Type:

Softsynth's five basic wave types: **Sine**, **Square**, **Triangle**, **Bandlimited noise** and **White noise**.

The **sine** wave is a "pure" wave because it is composed of only a single frequency. Sine waves do not have any additional harmonics, so they are ideal for controlled additive synthesis. You will probably want to use sine waves the most when working with Softsynth.

The **square** wave contains many harmonics above its fundamental frequency, resulting in a harsher, "buzzier" tone. These high harmonics can cause *aliasing* distortion (see Section 2), particularly if you use them for high frequency partials. Softsynth's square wave has been slightly bandlimited (high frequencies removed) to minimize aliasing problems. **However, use square waves carefully at high frequencies!**

The **triangle** wave also contains some high harmonics - follow the same precautions as with the square wave. If used judiciously, triangle waves and square waves can add more "bite" and "roughness" to a sound.

The **band-limited noise** wave type consists of filtered white noise with a bandwidth of approximately 500 Hz (centered around the partial's frequency). This wave type is useful for narrow band noise effects, such as simulating "breath" on a flute.

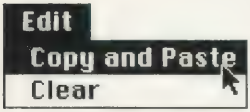
The **white noise** wave type is broad-band noise (equal amplitude at all audible frequencies). This wave type is useful for adding percussive attack to the start of sounds, and for synthesizing snare drum-like sounds.

Select a wave type by clicking on its icon. The wave type icons are located at the top of the Single Partial editing screen.

Editing Techniques

Single Partial editing provides very precise control over every aspect of each partial. The drawback of this technique is the time required to enter such a large number of "details" to create a sound. After you have spent some time working with Softsynth's Single Partial editing, you will learn some shortcuts and techniques that will speed up your sound creating efforts.

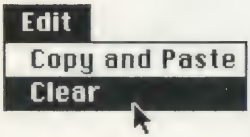
Editing Tools



Softsynth also provides several editing tools to help you create sounds faster.

The **Copy and Paste** command allows you to copy features of the current partial to one or more other partials. Any combination of the partial's amplitude envelope, frequency contour, wave type or ratio can be copied to a specified *range* of other partials.

To copy a partial's parameters and assign them to other partial(s), select **Copy and Paste** from the **Edit** menu. A dialog box will appear that allows you to select the parameter(s) of the current partial you wish to copy (using "checkboxes"), and a range of partials you want to transfer these parameters to. Once you have selected the parameters and entered a range of partials, click on the **Copy** button to perform the copy and paste. Click on the **Quit** button to abort the process.



The **Clear** function sets *selected* parameters to "zero" settings for the current partial. Clear removes all breakpoints from the amplitude envelope and/or tuning contour, returns the ratio to its default setting, and sets the wave type to sine wave.

Now that you understand the basic tools provided in Single Partial mode, spend some time experimenting. Create a few sounds and listen to the results.

There are two ways to synthesize and play back Softsynth sounds. The fastest way to listen to a sound you are working on is to synthesize it and play it out the Macintosh's internal speaker. This is called **Previewing** the sound.

Preview
(Mac speaker playback)

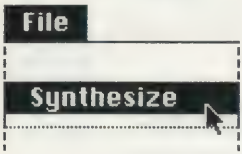
To preview a sound, click once on the speaker icon that appears in the main screen. After you release the mouse button, Softsynth will synthesize the sound. To play the sound out the Mac speaker, click on the speaker icon again (the sound will play each subsequent time you click on the speaker icon). If you make any changes to the sound's parameters, Softsynth will repeat the synthesis process before playing back the altered sound.

Single Partial Preview

If you click on the speaker icon that appears in the single partial screen, *only* the current partial will be synthesized and played.

Although the playback sound quality of the Mac is reasonably good, it is only 8 bit resolution. This sound quality can be greatly improved by connecting an external speaker and amplifier to the Mac's speaker output.

Synthesize

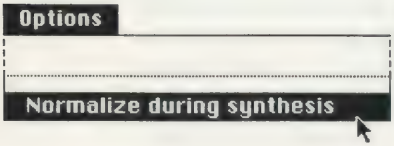


If you want to listen to a higher quality version of the sound you are creating, you must **Synthesize** it and transfer it to your sampler.

When Softsynth previews a sound, it synthesizes it with only 8 bit accuracy and does not save it on disk. When you have finished creating a sound (or just want to transfer it to your sampler), select **Synthesize** from the **File** menu.

Softsynth will then synthesize the sound with 16 bit accuracy, and save it to disk as a file of digital samples in *Sound Designer* format.

Normalize



You also have the option to **normalize** the overall amplitude of the sound when it is synthesized. Normalizing increases the sound's amplitude to the highest level possible without creating clipping distortion. You should always normalize sounds that you plan to use. However, the normalizing process takes an additional few seconds, so you may want to disable it when you are synthesizing "test" sounds or working on a sound. You can then turn normalizing on when you synthesize the final sound.

The normalize function can be switched on and off by selecting **Normalize** from the **Options** menu. Normalize is ON when a check mark appears next to its name in the options menu.

Transferring Sound Files to the Sampler

Once you have synthesized the sound, it can be transfered to your sampler using the **File->Sampler** command in the **File** menu. Section 5, "Sampler Personalities" includes transfer instructions for each sampler supported by Softsynth.

MIDI Keyboard

The sampler can be "played" from the Macintosh using Softsynth's *MIDI Keyboard*. This feature is very useful if you are using a rack-mount sampler without a MIDI keyboard controller. To display the MIDI keyboard, select **MIDI Keyboard** from the **Options** menu.

A window will appear displaying a 5-octave keyboard and 3 buttons. Click the cursor on the keyboard to manually "play" the sampler (MIDI data is transferred to the sampler to play the notes you click on). The MIDI Keyboard transmits and receives on MIDI channel 1 only.

The **Record** button allows you to record a short MIDI sequence. To record a sequence, click on the Record button and play the sampler's keyboard or Softsynth's MIDI Keyboard. Click anywhere in the MIDI Keyboard window to stop recording (or playback).

The **Play** button will play back the last recorded sequence. Sequences recorded by "playing" the Mac's keyboard will be played back in step time (each note has the same duration).

The **Pattern** button plays a slow glissando up the entire keyboard, allowing you to hear all sounds currently assigned to the keyboard.

Single Partial mode is Softsynth's most flexible editing mode - you have complete control over changes in the frequency and amplitude of each partial, as well as the overall harmonic content of the sound.

Time Slice Editing

However, Softsynth provides a second editing mode called **Time Slice** mode that may be easier or more intuitive for some people to use. Time Slice mode is particularly useful for creating a sound's basic structure - Single Partial mode can then be used to "fine tune" the sound.

Timbre Events

In Time Slice mode, a single envelope is provided that controls changes in the *overall* amplitude of the sound. The relative levels of the individual partials are controlled by **Timbre Events**. A timbre event is a "snapshot" of the amplitude of each partial at a single instant in time. The relative levels of all the partials is called a **Harmonic Mix**. A timbre event is a type of *breakpoint* that stores a harmonic mix.

Let's look at the Time Slice programming screen. If you are in Single Partial mode, click on the mixer icon to return to the main screen. Once the main screen appears, close the current parameter file (select Close from the File menu) and Open the parameter file titled "Time Slice demo". Click once on the **Time Slice** icon (a clock with a butcher knife). The time slice editing screen will appear on the Macintosh.

Note: You may see the message "Converting single partial data to time slice" briefly on the Mac screen after you click on the time slice icon. All parameters entered in single partial mode are "translated" into time slice mode, and vice versa.

Master Envelope

The top half of the time slice screen is a familiar sight. It is an envelope graph that works just like the single partial envelope graphs, with one important difference: the time slice envelope controls the sound's *overall* amplitude.

You can edit, create and delete breakpoints on the time slice envelope using the same techniques you learned on pages 24 - 25.

Timbre Event Bar

A long, narrow rectangle appears along the bottom of the time slice window. This is the **Timbre Event Bar**. It is about the same height as a scroll bar, and contains the timbre events. Each timbre event is represented by a tiny vertical bar icon. A number appears to the right of each timbre event icon. The numbers are assigned in the order the timbre events appear in the sound (the first timbre event at the start of the sound is number 1, the next is 2, etc.).

How does time slice mode create sounds? Softsynth *interpolates* (fades) between the individual timbre events to create a changing tonal quality (timbre). Each timbre event creates a breakpoint in each partial's envelope. The level of the partial at the timbre event's time is determined by the level of the partial's harmonic fader.

Once the sound's timbre has been created using the timbre events, changes in the sound's overall amplitude can be adjusted using the master envelope graph.

When you click on a timbre event icon, the harmonic faders display the *harmonic mix* stored by the timbre event.

Click once on the first timbre event (located at the left end of the timbre event bar). A harmonic mix will appear on the harmonic faders along the bottom of the screen.

The harmonic mix that appears on the faders should show partials 1, 2 and 3 at full amplitude (highest fader setting). All the other harmonic faders should be at their lowest settings (off).

Next, click on the second timbre event icon located about .25 seconds into the sound. The harmonic mix should show partials 4, 5 and 6 at full amplitude. Partials 1, 2 and 3 should be off.

So far, the sound starts with partials 1, 2 and 3 at equal levels. As partial 1-3 fade out, partials 4-6 fade in. The entire process occurs in only .25 second!

Creating Timbre Events

To better understand how time slice mode interpolates between timbre events, let's create a new timbre event. To create a timbre event, click the cursor in the timbre event bar in a spot that is *not* occupied by a timbre event. Create a new timbre event about halfway between timbre events 1 and 2.

A new timbre event will appear. Instead of being timbre event number 3, the new timbre event will become timbre event number 2 - the timbre event that *was* number 2 will become number 3. Remember, timbre events are numbered in the order they appear in the timbre event box, **not** the order they are created in.

The new timbre event is now selected - look at its harmonic mix. Partials 1 - 6 should all have about the same level. As partials 1-3 (from timbre event 1) are fading out, and partials 4-6 are fading in (towards timbre event 3) they are equal level halfway between (timbre event 2).

Moving Timbre Events

Next, drag timbre event 2 closer to timbre event 1. Timbre events can be moved by clicking on the timbre event icon, then dragging it to the left or right. When you release the mouse button, timbre event 2's new harmonic mix should show partials 1-3 higher than partials 4-6.

Experiment with creating and moving timbre events, and adjusting the master envelope. You can listen to the results of your experiments by returning to the main screen (click on the mixer icon), then using the **preview** function (click on the speaker icon).

We have now covered the basic techniques for editing sound parameters in both Single Partial and Time Slice modes. Section 6, "Tips and Tricks" discusses other techniques for creating interesting sounds using Softsynth.

Section 4

The Smartsynth Function

What is the Smartsynth Function?

Softsynth gives you precise control over all important additive synthesis parameters. However, sometimes you just want to quickly create a new type of sound without adjusting a lot of parameters and breakpoints. The **Smartsynth** function is an intelligent sound generator - it creates new sounds based on your general "description" of the sound you want.

How does it work?

Smartsynth uses *parameter ranges* and a *random number generator* to create new sounds. You select ranges for various Softsynth parameters using the Smartsynth **Setup Menu**, then Smartsynth creates a random value for each parameter within the range you have specified.

For example, if you select "Fast Attack" from the Setup menu, Smartsynth will generate a random attack time (elapsed time from the start of the sound to the partial's peak level) between 0 and 15 milliseconds for each partial. Select "Medium Attack" and each partial's attack will be a random value between 15 and 100 milliseconds.

Smartsynth also uses "models" of certain types of attack characteristics. The attack is a critical part of every sound, and these models are useful for creating various attack effects.

Smartsynth is *not* a substitute for learning additive synthesis techniques. You will get the best results from Softsynth if you take the time to learn additive synthesis and experiment with Softsynth's many parameters. Smartsynth is an excellent tool for generating new ideas and helping you explore new synthesis approaches.

Smartsynth will often create sounds that are interesting but not "refined" or "finished". Use Smartsynth to create the basic sound, then use Softsynth's single partial editing mode to refine and "polish" the sound.

Smartsynth can also be used to add new qualities to existing Softsynth sounds (parameter files). **Smartsynth will affect the current "open" parameter file - if you want to create a new sound, you must start with a new (blank) parameter file.**

Creating a New Sound

Let's create a new sound using Smartsynth. If you have a parameter file open, close it by selecting **Close** from the **File** menu. Next, select **Set up** from the pull-down **Smartsynth** menu.

Parameter Types

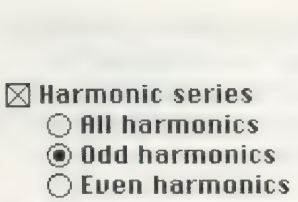
The Smartsynth Setup menu will appear. This menu contains the Smartsynth parameters. They are organized in two basic groups: **Timbre** parameters and **Envelope** parameters.

The **Timbre** parameters primarily affect the frequency characteristics of each partial, including harmonic series, ratio and frequency contour.

The **Envelope** parameters primarily affect the amplitude characteristics (envelope) of each partial. The *Frequency Attack* parameter is an exception - it is a frequency parameter included with the envelope parameters.

To activate a parameter, click in the check box next to the parameter's name. A parameter is active when an "X" appears in the check box.

Let's look at the Timbre parameters first.



The **Harmonic Series** parameter determines which partials Smartsynth will use. This parameter must be active (on) for Smartsynth to create a new sound. Smartsynth will select the basic harmonic series from the first 16 harmonics - the other 16 harmonics (17-32) are reserved for doubling and special effects.

Click once on the button next to the desired harmonic series:

All Harmonics - Smartsynth will select any harmonics from 1 through 16 to create the sound's harmonic series.

Odd Harmonics - Smartsynth will select any *odd* numbered (3, 5, 7, etc.) harmonics from 1 through 16 to create the sound's harmonic series.

Even Harmonics - Smartsynth will select any *even* numbered (2, 4, 6, etc.) harmonics from 1 through 16 to create the sound's harmonic series.

The harmonic series will always include the fundamental (partial 1).

☒ Harmonic range
- ○ High
- ◉ Medium
- ○ Low

The **Harmonic Range** parameter determines the highest partial (from 1-16) that will be included in the sound's harmonic series. The range of available partials includes all partials *below* the highest partial.

High - the sound's highest partial will be selected from partials 12 - 16.

Medium - the sound's highest partial will be selected from partials 8 - 11.

Low - the sound's highest partial will be selected from partials 3 - 7.

This parameter is useful for limiting the number of higher partials in a sound. For example, if you want a pure "bass" sound (few high harmonics), select **Low**.

If the Harmonic range parameter is not active (off), the harmonic series will not be affected.

☒ Harmonic filter
- ◉ Slight
- ○ Medium
- ○ Extreme

The **Harmonic Filter** parameter lowers the fader levels of higher harmonics - it "filters" higher harmonics from the sound without completely removing them (some low level high partials may be removed).

The Harmonic Filter has three settings: **High**, **Medium** and **Low**. The **Low** setting slightly lowers the fader levels of high partials. The **High** setting drastically lowers the fader settings of high partials.

Many sounds that contain high level upper partials sound very bright and piercing. Use the Harmonic Filter parameter to reduce the level of the higher partials for a more "natural" tonal balance.

☒ Partial detuning
○ Slight
◉ Medium
○ Extreme

The **Partial Detuning** parameter changes the ratio of each partial from 1 -16 to a "non-standard" value. For example, the ratio of partial 5 might be changed to 4.95, partial 6 to 6.03, etc.

Each partial will be changed by a different amount. The range of change allowed is determined by the parameter setting:

Slight - The ratio of each partial will be changed by less than 1%. This setting will cause a subtle change in the sound's timbre.

Medium - The ratio of each partial will be changed by less than 2%. This setting will cause a slightly more pronounced change in the sound's timbre.

Extreme - The ratio of each partial will be changed by less than 50%. This setting will create a very enharmonic harmonic series, usually producing metallic or bell-like sounds.

☒ Doubling
◉ Slight
○ Medium
○ Extreme

The **Doubling** parameter copies all parameters of partials 1 - 16 to partials 17 - 32, then detunes the ratios of partials 17 - 32. These detuned partials are mixed at the same levels as the standard partials to create **detuned pairs** of partials.

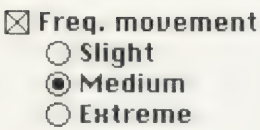
For example, if a sound contains partials 2, 3 and 5 (each partial has the standard ratio), then the Doubling parameter will create three corresponding detuned partials such as 1.98, 2.97 and 4.95. All of the partials will be detuned by the same *relative* amount (i.e. 1%).

The frequency detuning range is controlled by the **Slight**, **Medium** and **Extreme** buttons.

Slight - The ratios of partials 17 - 32 will be within .5% of the ratios of partials 1 - 16. This setting creates a slight doubling effect, making the sound "fatter".

Medium - The ratios of partials 17 - 32 will be within 1.5% of the ratios of partials 1 - 16. This setting creates a more pronounced doubling effect.

Extreme - The ratios of partials 17 - 32 will be within 3% of the ratios of partials 1- 16. This setting creates a very strong doubling effect.



The **Frequency Movement** parameter inserts breakpoints in the tuning contour graph of each partial. This causes the pitch of each partial to fluctuate (go sharp and flat) during the sound. This parameter adds "motion" to the sound, and creates a chorusing effect.

The number of breakpoints inserted in the frequency contour of each partial is random. The pitch change range is controlled by the parameter setting:

Slight - The pitch of each partial changes by less than .5% (sharp or flat). This setting creates a subtle chorusing effect.

Medium - The pitch of each partial changes by .5% to 1% (sharp or flat). The chorusing effect is more pronounced.

Extreme - The pitch of each partial changes by 1% to 2% (sharp or flat). This setting creates audible pitch changes.

The **Envelope** parameters are displayed in the lower half of the Smartsynth menu.

☒ Attack rate
◉ Fast
○ Medium
○ Slow

The **Attack Rate** parameter controls the attack time of the sound. The attack time is determined by the position of the first amplitude breakpoint of each partial:

Fast - The first amplitude breakpoint of each partial occurs within the first 15 milliseconds of the sound.

Medium - The first amplitude breakpoint of each partial occurs within the first 75 milliseconds of the sound.

Slow - The first amplitude breakpoint of each partial occurs within the first 250 milliseconds of the sound.

☒ Frequency attack
◉ Bend
○ Bite
○ Blip

The **Frequency Attack** parameter consists of three "modeled" attack tuning contours. When the Frequency Attack parameter is active (on), the selected attack model is inserted at the beginning of each partial's tuning contour. The attack models include **Bend**, **Bite** and **Blip**. Each attack model has a characteristic sound that is difficult to describe. Listen to the sounds, and look at the start of the tuning contour graphs produced by each type of frequency attack.

The *Frequency Attack* and *Frequency Movement* parameters can not be used in the same sound. The Frequency Movement parameter requires a 2% tuning contour range, and the Frequency Attack parameter requires a 50% tuning contour range. Only one tuning contour range setting can be used in a parameter file. Selecting either of these parameters will automatically "turn off" the other parameter.

☒ Percussion
○ Dink
○ Bump
◉ Click

The **Percussion** parameter uses partial 32 to add a percussive attack effect to the beginning of the sound.

Dink - Creates a metallic "clink" attack using a high frequency (2,000 - 10,000 Hz) sine, square or triangle wave and an envelope with a fast decay.

Bump - Creates a "thump" sound using a low frequency sine wave (ratio = .20 to .5) and an envelope with a fast decay.

Click -Creates a "knock" sound using mid frequency bandlimited noise (800 to 4,000 Hz) and an envelope with a fast decay.

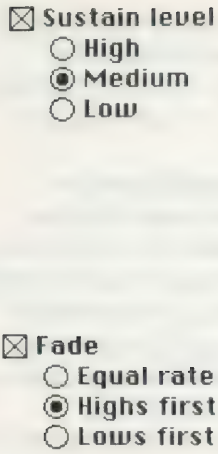
☒ Decay rate
○ Fast
◉ Medium
○ Slow

The **Decay Rate** parameter controls the initial decay time of the sound. The decay time is determined by the position of the second amplitude breakpoint of each partial:

Fast - The second amplitude breakpoint of each partial occurs within 15 milliseconds of the first (attack) breakpoint.

Medium - The second amplitude breakpoint of each partial occurs within 75 milliseconds of the first (attack) breakpoint.

Slow -The second amplitude breakpoint of each partial occurs within 250 milliseconds of the first (attack) breakpoint.



The **Sustain Level** parameter controls the general sustain level of the sound after the attack and decay. The overall sustain level is controlled by the level of the second amplitude breakpoint of each partial.

Once a sustain level is set, Smartsynth limits the amount each partial's amplitude can deviate from the selected sustain level.

The **Fade** parameter causes the sound to "fade out" to silence at its end.

Equal Rate - All partials will fade at approximately the same rate.

Highs First - High partials will fade faster than low partials.

Lows First - Low partials will fade faster than high partials.

Using Smartsynth to Modify an Existing Sound

The Smartsynth function can also be used to modify an existing parameter file. Any Smartsynth parameter can be independently "applied" to a sound you are working on.

However, any time Smartsynth creates breakpoints or ratios, it will overwrite the original breakpoints or ratios. For example, if you select Smartsynth's *Frequency Movement* function, all existing tuning contour breakpoints will be removed from the partials, and replaced by Smartsynth's new breakpoints. Always make a backup copy of the parameter file *before* you use Smartsynth to modify it.

An existing parameter file must be **open** to be "processed" using Smartsynth. Select the parameters you want to process using the Smartsynth Set Up menu. Select *only* the parameters you want to change - make sure the other parameter boxes are *not* selected (no check marks in their boxes).

Once you have set the parameters you want to process, click on the **OK** button to return to the main screen. Each time you select **Generate** from the **Smartsynth** menu, Smartsynth will create new values for each selected parameter.

For example, if you selected the *frequency movement* parameter, new tuning contour breakpoints will be created for *every* partial each time you select **Generate**.

Section 5

Sampler Personalities

About Samplers

Digital sampling instruments are really digital audio recorders that can transpose (raise or lower) the pitch of the digital recording.

Softsynth creates simulated digital recordings ("samples") of sounds you have designed using the programming screens discussed in Section 3, "Using Softsynth". Once you have created a sound using Softsynth, it must be transferred to your sampler for playback. Softsynth can create sounds for several different samplers, and each sampler has a different "personality".

Fortunately, Softsynth already knows all about your sampler's personality. After you select your sampler from the **Sampler** menu, Softsynth's **File->Sampler** function will automatically convert Softsynth sounds into the correct format for your sampler.

All samplers are not created equal - many factors contribute to the overall fidelity of a sampler, including *sample rate*, *data format* and *pitch shifting* method.

The **sample rate** is the number of times each second that the sampler measures the input sound. Increasing the sample rate increases the high frequency response. In theory, the highest frequency a sampler can record and reproduce is one half the sample rate. For example, if the sample rate is 30,000 samples-per-second, the highest reproducible frequency would be 15,000 Hertz. Actually, the real high frequency limit is slightly lower due to input and output filtering.

Don't create partials with frequencies higher than one half the sample rate. If you do, Softsynth will display a dialog box warning you that the partial is "out of range".

Some samplers have variable sample rates. If your sampler has more than one sample rate, the sampler should be set to the same sample rate as Softsynth's main programming screen. If the selected sample rates are different, the Softsynth sound may play back at the wrong pitch.

The **data format** is the digital recording method used by the sampler to record and store sounds. In general, the more bits used to record each sample, the better the signal-to-noise ratio and dynamic range. 8 bit *linear* (no digital compression) sampling provides a maximum signal-to-noise ratio of 48 decibels (dB), 12 bit linear sampling has a S/N ratio of 72 dB, and 16 bit linear sampling has a S/N ratio of 96 dB. Some samplers (such as the E-mu Systems Emulator II and Emax) use various data compression techniques and other tricks to improve their signal-to-noise ratio.

Softsynth creates 16 bit linear sound files in standard Sound Designer format. These files are converted into your sampler's format when the **File->Sampler** function is used.

Each sampler also uses a slightly different method for transferring sound files to and from the Macintosh. The following descriptions cover both the sampling and sound transfer characteristics of Softsynth compatible samplers.

Prophet 2000/2002

The Sequential Prophet 2000 is a twelve bit sampler with a maximum sampling time ranging from 6 to 16 seconds, depending on sample rate (up to 32 seconds with an optional memory upgrade). Softsynth transfers sounds to the Prophet 2000 at twice the normal MIDI rate - 63,000 bits per second.

Sample Rates

The Prophet 2000 has three sample rates: 15625 Hz, 31250 Hz and 41667 Hz. 31250 is the most commonly used sample rate, so Softsynth automatically enters 31250 in the main screen *Sample Rate* parameter box. If you plan to use either of the other Prophet 2000 sample rates, be sure to enter the correct value in the *Sample Rate* box.

Transferring sound files to the Prophet

You must *synthesize* a Softsynth sound before you can transfer it to the Prophet. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Once you have synthesized the sound, select **File->Sampler** from the **File** menu.

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the Prophet, then click once on the **Open** button.

If the Macintosh is not able to communicate with the Prophet for any reason (e.g. the MIDI interface is not connected, the wrong serial port is selected, etc.), a "Can't Communicate" alert box will appear. If this happens, check the MIDI interface and serial port settings. Be sure the Prophet is *not* in "Record Sample" mode.

A dialog box listing all of the Prophet 2000's samples and their sizes (in number of samples) will appear. Select a sample number that is labeled "Not Sampled" and click on the **OK** button. If want to replace an existing sample with the Softsynth sound, you must first *delete* the existing sample from the Prophet front panel.

Sound Length

If the Prophet 2000 does not have enough free sound memory to store the Softsynth sound, a dialog box will appear to warn you that the Softsynth file will be truncated (cut-off).

Mirage/ Multisampler

The Ensoniq Mirage and the Mirage Multisampler are both 8 bit samplers that can store about 4.4 seconds of digitized sound (2.2 seconds per memory half) at a sample rate of 29412 Hz. Softsynth transfers sound files to both Mirage models at the standard MIDI baud rate (31.5k). The transfer procedure is identical for both models.

Sample Rates

The Mirage and Multisampler have variable sample rates with many possible settings. Softsynth enters a default sample rate of 29412 Hz in the main screen *Sample Rate* parameter box. This is the Mirage's default playback sample rate. If you use a different sample rate, be sure you enter the correct value in the *Sample Rate* box.

Transferring sound files to the Mirage

You must *synthesize* a Softsynth sound before you can transfer it to the Mirage. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Once you have synthesized the sound, select **File->Sampler** from the **File** menu.

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the Mirage, then click once on the **Open** button.

If the Macintosh is not able to communicate with the Mirage for any reason (e.g. the MIDI interface is not connected, the wrong serial port is selected, etc.), a "Can't Communicate" alert box will appear. If this happens, check the MIDI interface and serial port settings.

A dialog box listing all of the Mirage's wavesamples and their sizes (in number of samples) will appear. Use the **Upper** and **Lower** buttons to select the memory half you want to transfer the Softsynth sound to. Select a wavesample and click on the **OK** button.

Memory Map

A "memory map" of the Mirage's sound memory appears at the bottom of the **File->Sampler** dialog box. The horizontal bar represents the Mirage's total sound memory, and shows the amount of memory used by each wavesample.

Sample Length

If the Mirage does not have enough free sound memory to store the Softsynth sound, a dialog box will appear to warn you that the Softsynth file will be truncated (cut-off). You have three options - you can **Cancel** the transfer, **Truncate** the file, or **Extend** the sample's memory. The Extend option may overwrite other data stored in the Mirage's memory, replacing or erasing part of another sample.

Emulator II

The E-mu Systems Emulator II uses a special non-linear 8 bit data format that provides a signal-to-noise ratio comparable to 12-14 bit systems. The Emulator II has a total sampling time of 17.2 seconds at a sample rate of 27777 kHz.

Sound Designer™ required

To use Softsynth with the Emulator II you must have the Emulator II version of our Sound Designer software. Why? Softsynth transfers sound files to the Emulator II using either of the Macintosh RS-422 serial ports (the **Printer** or **Modem** port). The RS-422 interface transfers sound files very fast - nearly 17 times MIDI speed (500,000 bits ber second). However, the Emulator can't receive sound files over MIDI. The only way to transfer sound files to the Emulator is via RS-422, so you must have the RS-422 interface cable provided with Sound Designer.

Connect the Macintosh and Emulator II following the instructions provided in the Sound Designer manual. This setup will also work for Softsynth. Select the serial port that the interface cable is connected to using the MIDI Setup dialog box (select **MIDI Setup** from the **Sampler** menu).

Sample Rate

The Emulator II uses a fixed sample rate of 27777 kHz. Softsynth automatically enters this value in the main screen *Sample Rate* parameter box when **Emulator II** is selected from the **Sampler** menu. Always use this sample rate when creating sounds for the Emulator II. If you use a different sample rate, the sound will play back at the wrong pitch on the Emulator.

Transferring sound files to the Emulator

You must *synthesize* a Softsynth sound before you can transfer it to the Emulator. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Once you have synthesized the sound, select **File->Sampler** from the **File** menu.

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the Emulator, then click once on the **Open** button.

If the Macintosh is not able to communicate with the Emulator for any reason, a "Can't Communicate" alert box will appear. If this happens, check the interface cable and serial port settings.

A dialog box listing all of the Emulator's voices (samples) and their sizes (in number of samples) will appear. The Softsynth sound file must replace an existing Emulator II voice. The Emulator II voice you select to replace must be *at least* the same length as the Softsynth sound file or longer.

Select the voice you want to replace and click on the **OK** button.

Sound Length

If the Softsynth sound is longer than the selected Emulator voice, a "Sound File Too Long" alert box will appear. The alert box will ask you if you want to truncate the sound (shorten it to fit the available memory space). If you choose to not truncate the sound, the transfer will be cancelled.

Korg DSS-1

The Korg DSS-1 is a twelve bit sampler with a maximum sampling times of 5.5 seconds (48 kHz sample rate) to 16 seconds (16 kHz sample rate). Softsynth transfers sounds to the DSS-1 at the standard MIDI rate - 31,500 bits per second.

Sample Rates

The DSS-1 has four sample rates: 48 kHz, 32 kHz, 24 kHz and 16 kHz. 32 kHz is the most commonly used sample rate, so Softsynth automatically enters 32000 in the main screen *Sample Rate* parameter box (remember, kHz stands for *thousands* of Hertz). If you plan to use any of the other DSS-1 sample rates, be sure to enter the correct value in the *Sample Rate* box.

You must *synthesize* a Softsynth sound before you can transfer it to the DSS-1. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Once you have synthesized the sound, select **File->Sampler** from the **File** menu.

Transferring sound files to the DSS-1

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the DSS-1, then click once on the **Open** button.

If the Macintosh is not able to communicate with the DSS-1 for any reason (e.g. the MIDI interface is not connected, the wrong serial port is selected, etc.), a "Can't Communicate" alert box will appear. If this happens, check the MIDI interface and serial port settings.

A dialog box listing all of the samples stored in the DSS-1and their sizes (in number of samples) will appear. Select a sample you want to replace and click on the **OK** button.

The DSS-1 organizes samples in groups. Each group is called a "multisound", and can contain up to 16 samples. The samples are listed by the name of their multisound followed by a dash and the sample number. For example, "Piano - 3" is the third sample in the multisound "Piano".

Sound Length

If the DSS-1 does not have enough free sound memory to store the Softsynth sound, a dialog box will appear to warn you that the Softsynth file will be truncated (cut-off). You have three options - you can **Cancel** the transfer, **Truncate** the file, or **Extend** the sample's memory. The Extend option may overwrite other data stored in the DSS-1's memory, replacing or erasing part of another sample.

Akai S900

The Akai S900 is a twelve bit sampler with maximum sampling times from 11.75 seconds (40 kHz sample rate) to 63.3 seconds (7.5 kHz sample rate). Softsynth transfers sounds to the S900 at the standard MIDI rate - 31,250 bits per second.

Sample Rates

The S900 has a *variable* input sample rate that has many possible settings ranging from 7.5 kHz to 40 kHz. 31250 is a commonly used sample rate, so Softsynth automatically enters 31250 in the main screen *Sample Rate* parameter box. If you create a Softsynth sound using a different sample rate, Softsynth will automatically set the S900's playback sample rate to match the Softsynth sample rate.

Transferring Sound Files to the S900

You must *synthesize* a Softsynth sound before you can transfer it to the S900. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Unlike other samplers, Softsynth cannot replace individual sounds in the S900's memory. Every time you transfer a sound to the S900 it will be stored in the S900's memory as sample number 00. To create a *program* of multiple samples, you must first save each Softsynth sound file to the S900's disk after you transfer it from the Macintosh.

Before transferring Softsynth sounds to the S900, "clear" the S900's memory by switching it off, then on with **no disk** in the disk drive.

Next, select **File->Sampler** from the **File** menu.

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the S900, then click once on the **Open** button.

If the Macintosh is not able to communicate with the S900 for any reason (e.g. the MIDI interface is not connected, the wrong serial port is selected, etc.), a "Can't Communicate" alert box will appear. If this happens, check the MIDI interface and serial port settings.

Push the **Disk** button on the S900 and select **Page 6** ("Save Sample"). The name of the sound file you transferred will appear as the current sample. Push the S900 "right arrow" once to advance the cursor to the sample, then push the S900 **ENT** button to save the sample to disk.

Repeat this process to transfer up to 32 Softsynth files to the S900. Remember, you **must** save each sound file to the S900's disk before transferring another sound file from Softsynth.

After you have transferred all of the desired Softsynth sound files, you can load the samples from the S900's disk, then use the S900's **Edit Program** functions to assign them to the keyboard and create programs.

If the S900 runs out of sample memory during a sound file transfer, the transferred sound will be truncated (cut off).

Akai S612

TheAkai S612 is a twelve bit sampler with a maximum sampling times from 1 second (32 kHz sample rate) to 8 seconds (4 kHz sample rate). Softsynth transfers sounds to the DSS-1 at the standard MIDI rate - 31,500 bits per second.

Sample Rates

The S612 has four primary sample rates: 4 kHz, 8 kHz, 16 kHz and 32 kHz. Intermediate sample rates can also be selected (see the S612 manual for more information). 16 kHz is the most commonly used sample rate, so Softsynth automatically enters 16000 in the main screen *Sample Rate* parameter box. If you plan to use any of the other S612 sample rates, be sure to enter the correct value in the *Sample Rate* box.

Transferring sound files to the S612

You must *synthesize* a Softsynth sound before you can transfer it to the S612. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Once you have synthesized the sound, select **File->Sampler** from the **File** menu.

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the S612, then click once on the **Open** button.

If the Macintosh is not able to communicate with the S612 for any reason (e.g. the MIDI interface is not connected, the wrong serial port is selected, etc.), a "Can't Communicate" alert box will appear. If this happens, check the MIDI interface and serial port settings.

Softsynth will transfer the sound file to the S612, completely erasing any sound that is already in the S612's memory.

Sound Length

If the S612 does not have enough free sound memory to store the Softsynth sound, a dialog box will appear to warn you that the Softsynth file will be truncated (cut-off). If this is the case, you may want to synthesize the sound at a lower sample rate to provide a longer sample time.

Section 6

Tips and Tricks

Start with a Parameter File Copy

Why re-invent the wheel each time you create a new sound? It can take some time to type in non-standard ratios, create a lot of breakpoints, etc. Try starting with a copy of an existing parameter file, then modify various parameters to change the sound. Experiment with the parameter files provided on the Softsynth parameter files disk.

To make a copy of an existing parameter file, select the file's icon while in the Macintosh Finder (the "desktop"), hold down the command key and type a "D". This will create an exact copy of the parameter file. Rename the file, open it, and you are ready to modify its parameters.

The *Copy and Paste* and *Clear* functions can be very useful for making overall changes to the parameter file.

Faster Synthesis

Softsynth performs thousands of mathematical calculations each time you synthesize or preview a sound. All this math takes time, and can be frustrating when you are creating a long or complex sound. Fortunately, there are a few things you can do to speed the synthesis process:

Shorten the length of a sound while you are working on it, then increase the length before you synthesize the final sound file. The shorter the sound, the fewer samples Softsynth has to calculate. Shorter sounds can be synthesized much faster.

Don't add unneccessary partials to the sound. The fewer partials included in a sound, the faster it can be synthesized. Don't assume that you can only create interesting sounds by using many partials. You can usually get

interesting results with just a few partials if you take advantage of independant envelopes and tuning contours.

Try to make edits in groups. Instead of making a single edit and then listening to the change, make a few edits. Think of the different qualities you want in your sound, then make a series of edits to create them.

Creating Presets

After you've created a sound that you like, you'll probably want to create a preset on your sampler with the sound spread across the entire keyboard. You should synthesize at least 4 samples at different frequencies to minimize the undesirable side effects of extreme pitch shifting.

If you only change the *Frequency* parameter for each different sample, you may find that your preset sounds bright in the high end and dull in the low end. This is because the higher harmonics of most musical instruments "roll-off" as the fundamental pitch rises. You can compensate for this effect by boosting the fader levels of the higher partials of low frequency samples, and decreasing the fader levels of the higher partials of high frequency samples.

Velocity Effects

If your sampler has any velocity crossfade or velocity switching capabilities, take advantage of them! Create two different versions of each sample - one should be brighter with a harder attack, and another should be softer and mellower. Adjusting the harmonic mix and the attack time of each partial can create interesting "velocity" effects.

Sound Designer™ Tips

Sound Designer™ is a powerful waveform editing and digital signal processing program we offer for many of the Softsynth compatible samplers. Using Softsynth and Sound Designer, you can create sounds, then process, edit, combine and modify them in many ways. The following tips are for Sound Designer users.

Crossfade Looping Detuned Sounds

If you create a doubled sound (i.e. you duplicate and detune a set of partials) it will probably be difficult to loop. If you own Sound Designer, you can use crossfade looping to quickly create a good loop. Crossfade looping works best when your loop markers are not close to the beginning or end of the sound file. You may want to synthesize a longer sound file when using doubling. You can then crossfade loop the sound, and then truncate the end using Sound Designer.

Using Frequency Analysis

Another powerful feature of Sound Designer is its Fast Fourier Transform (FFT) frequency analysis. Sound Designer's FFT divides the sound into 128 separate frequency bands, then displays a high-resolution, 3-D surface that shows both the frequency content of the sound, as well as changes in the amplitude of each frequency as time passes. In most cases, the harmonics of a sound can be easily seen. This is the Sound's "blueprint", and it is the same information that you create when you design a Softsynth sound.

Sound Designer can be a very useful tool for analysing and re-creating acoustic sounds with Softsynth. Of course, there's no point in painstakingly reconstructing an exact duplicate of a sampled sound. However, with frequency analysis and Softsynth you typically don't get exactly what you started with.

The results often have the complexity of acoustic sounds, along with a certain synthetic quality. Another difference between sampled sounds and Softsynth sounds is that Softsynth sounds tend to be cleaner and can easily be made brighter to compensate for the loss of high frequencies that occurs during sampling.

Combining Softsynth and Karplus-Strong

You can also get some interesting results by combining Softsynth's additive synthesis with Sound Designer's Karplus-Strong synthesis. Synthesize a sound with a lot of timbral movement using time slice editing and save it to disk. Then use this synthesized sound as the input sound file to the resonator in Sound Designer's Karplus-Strong module. Be sure to experiment with resonator percentages - lower numbers produce more Karplus-Strong effect.

Section 7

Case Studies

Programming Style

Every Softsynth user will develop his or her own "style" of creating sounds - some will start in time slice mode then adjust each individual partial in single partial mode, others will work entirely in single partial mode, and some may prefer to just sit back and let Smartsynth do the work. Of course, Softsynth users can improve their programming skills by sharing tips and techniques with each other. That is the purpose of the "Case Studies" included in this section of the manual.

A case study is a written description of the techniques used to create a Softsynth sound. A sample case studies is included in this section that describes the creation of one parameter file on the Softsynth parameter files disk.

Send Us your Case Studies!

However, the main purpose of this section is to encourage you to write some case studies and share them with other Softsynth users. Here's how you can participate:

Create a Softsynth sound and write a detailed description of the techniques you used. Send a disk that contains the written description (preferably in a Macwrite document) and a copy of the parameter file to Digidesign, along with your name, address and phone number.

We will select the best case studies and publish them in our newsletter, as well as distributing copies of the description freely to Softsynth users. If we use one of your case studies, we'll send you a check for $50 and a "goodie". This is your big chance for fame and fortune!

ClarinetMJ -
A Case Study by Mark Jeffery

For "ClarinetMJ" I used the Sound Designer frequency analysis (FFT) approach. The original clarinet sample that I analyzed came from an Emulator II disk. There were six samples on the disk that I could choose from for the basis of my Softsynth sound. Of the six clarinet samples the highest two lacked personality. They weren't "obvious" clarinets so I ruled them out for being my source sample. The lower four samples all sounded very good, so by default I selected the lowest sample for frequency analysis.

This was a mistake. Since the fundamental frequency of the lowest sample was probably around 150 Hz its harmonics were at 150 Hz, 300 Hz, 450 Hz, 600 Hz etc. Since frequency analysis is done at 100 Hz intervals (i.e 100 Hz, 200 Hz, 300 Hz, 400 Hz etc.) it was very difficult to discern harmonics. All the frequency plot could tell me was that the sound energy was closely grouped together.

Learning from my mistake I selected sample #3 as my source sample.
Its fundamental frequency was around 500 Hz. Since its harmonics were at 500 Hz, 1000 Hz, 1500 Hz etc., they were much easier to see. Ridges (i.e harmonics) were quite apparent.

The next step was to figure out what each harmonic's envelope was. To do this I set up the frequency analysis to do short (50 ms) segments of the sound. By scrolling through the sound using the time axis scroll bar I did a series of FFT's at different points in time. Each FFT was printed out using my Imagewriter by hitting <command, shift, '4'> (this a Macintosh feature that apparently is not widely known). I also printed out an FFT of the entire sound (350 ms).

With a pile of printouts by my side, I then started up Softsynth. The first thing I did was set the faders for each harmonic. I did this by deducing the relative amplitude of each harmonic from the FFT of the entire sound. The loudest harmonic received a fader level of 63 and the other harmonics were set relative to it. Sound energy that appeared as enharmonic was excluded from the sound since it was probably some type of extraneous noise. Next, I created the amplitude envelopes for each harmonic. Since the relative amplitude of each harmonic had already been set with the faders, each envelope was normalized (i.e. its peak value equals 100%). Break points were determined by looking for amplitude changes in a harmonic. I recreated the general shape of each envelope and ignored the slight amplitude fluctuations which are inherent in acoustic sounds.

Since the FFT can't show the slight frequency fluctuations which are also inherent in acoustic sounds I excluded frequency break points from the sound. I figured I could try to improvise with frequency break points later if I found it necessary.

Next came the moment of truth. I played my sound out of the Macintosh speaker. I was a little disappointed. But, knowing full well that the fidelity of the Mac is not great, I created a sound file and transfered it to my Prophet 2000 for playback. It sounded much better. I also noticed that it was much cleaner than the original sample although it also lost a little bit of personality. I played around with the harmonic mix in order to give the sound a little more edge. After tweaking some envelopes I was satisfied with it.

Next came the second moment of truth. I did a frequency analysis of my Softsynth clarinet. Did it look just like the original sample? Not exactly. Did I remain calm because I still had created a musically useful sound? Yes.

Hard Disk Installation

Softsynth's copy protection allows you to install the program on a hard disk without requiring you to insert the master disk when you start the program.

Never delete the Softsynth program icon from a hard disk! You must use the *Remove* function for Softsynth to be re-installed on a hard disk.

To install Softsynth on a hard disk:

1) Boot the Macintosh from the hard disk.

2) Insert the original Softsynth master disk (a copy won't work). The disk must be unlocked. Start Softsynth from this disk by double-clicking on the Softsynth icon.

3) A hard disk install/remove menu will appear. Click on the **Install** button.

4) A dialog box will appear that lists the available disk volumes and folders that Softsynth can be installed on. Use the **Disk** button to select the hard disk or hard disk volume to install Softsynth on. You cannot move Softsynth to a different volume once it has been installed.

5) Click once on the **Install** button. After the installation is complete the hard disk install/remove menu will appear. Click on the **Execute Installed** button to run Softsynth, or the **Finder** button to quit to the Finder.

To Remove Softsynth from a hard disk:

1) Boot the Macintosh from the hard disk.

2) Insert the original Softsynth master disk (a copy won't work). The disk must be unlocked. Start Softsynth from this disk by double-clicking on the Softsynth icon.

3) A hard disk install/remove menu will appear. Click on the **Remove** button.

4) A dialog box will appear that lists available disk volumes and folders. Use the **Disk** button to select the hard disk or hard disk volume that Softsynth is installed on. Locate the Softsynth file and select it (click on its name).

5) Click once on the **Remove** button. After the remove is complete, the hard disk install/remove menu will appear. Read the dialog box: it should say that "you have the option to install 1 copy...". Click on the **Finder** button to quit to the Finder.

NOTE TO FREQUENCY CONVERSION CHART (C4 = MIDDLE C; A4 = 440HZ)

note	frequency (Hz)	note	frequency (Hz)	note	frequency (Hz)
C0	16.35 Hz	C3	130.81 Hz	C6	1046.50 Hz
C0#	17.32 Hz	C3#	138.59 Hz	C6#	1108.73 Hz
D0	18.35 Hz	D3	146.83 Hz	D6	1174.66 Hz
D0#	19.45 Hz	D3#	155.56 Hz	D6#	1244.51 Hz
E0	20.60 Hz	E3	164.81 Hz	E6	1318.51 Hz
F0	21.83 Hz	F3	174.61 Hz	F6	1396.91 Hz
F0#	23.12 Hz	F3#	185.00 Hz	F6#	1474.98 Hz
G0	24.50 Hz	G3	196.00 Hz	G6	1567.98 Hz
G0#	25.96 Hz	G3#	207.65 Hz	G6#	1661.22 Hz
A0	27.50 Hz	A3	220.00 Hz	A6	1760.00 Hz
A0#	29.14 Hz	A3#	233.08 Hz	A6#	1864.66 Hz
B0	30.87 Hz	B3	246.94 Hz	B6	1975.53 Hz
C1	32.70 Hz	C4	261.63 Hz	C7	2093.01 Hz
C1#	34.65 Hz	C4#	277.18 Hz	C7#	2217.46 Hz
D1	36.71 Hz	D4	293.67 Hz	D7	2349.32 Hz
D1#	38.91 Hz	D4#	311.13 Hz	D7#	2489.02 Hz
E1	41.20 Hz	E4	329.63 Hz	E7	2637.02 Hz
F1	43.65 Hz	F4	349.23 Hz	F7	2793.83 Hz
F1#	46.25 Hz	F4#	369.99 Hz	F7#	2959.96 Hz
G1	49.00 Hz	G4	392.00 Hz	G7	3135.96 Hz
G1#	51.91 Hz	G4#	415.31 Hz	G7#	3322.44 Hz
A1	55.00 Hz	**A4**	**440.00 Hz**	A7	3520.00 Hz
A1#	58.27 Hz	A4#	466.16 Hz	A7#	3729.31 Hz
B1	61.74 Hz	B4	493.88 Hz	B7	3951.07 Hz
C2	65.41 Hz	C5	523.25 Hz	C8	4186.01 Hz
C2#	69.30 Hz	C5#	554.37 Hz	C8#	4434.92 Hz
D2	73.42 Hz	D5	587.33 Hz	D8	4698.64 Hz
D2#	77.78 Hz	D5#	622.25 Hz	D8#	4978.03 Hz
E2	82.41 Hz	E5	659.26 Hz	E8	5274.04 Hz
F2	87.31 Hz	F5	698.46 Hz	F8	5587.65 Hz
F2#	92.50 Hz	F5#	739.99 Hz	F8#	5919.91 Hz
G2	98.00 Hz	G5	783.99 Hz	G8	6271.93 Hz
G2#	103.83 Hz	G5#	830.61 Hz	G8#	6644.88 Hz
A2	110.00 Hz	A5	880.00 Hz	A8	7040.00 Hz
A2#	116.54 Hz	A5#	932.33 Hz	A8#	7458.62 Hz
B2	123.47 Hz	B5	987.77 Hz	B8	7902.13 Hz

Softsynth Credits

Softsynth Concept/Functional Design:
Mark Jeffery, Peter Gotcher, Evan Brooks

Software:
Mark Jeffery, Evan Brooks

Manual:
Peter Gotcher

Special thanks to:
Dana Massie, everyone at Digidesign

Emax

The E-mu Systems Emax uses a special non-linear 8 bit data format that provides a signal-to-noise ratio comparable to 12-14 bit systems. The Emax has several sample rates ranging from 10 kHz to 42 kHz. Total sampling time is 17.2 seconds at the default sample rate of 27777 kHz.

Softsynth can transfer sound files to the Emax using either a MIDI Interface or an optional RS–422 interface cable (available from Digidesign). This interface cable is included with the Emax version of Sound Designer™ - if you own the Emax Sound Designer you do not need to purchase another interface cable. The primary advantage of the RS-422 interface is speed - sound files are transferred at a rate of 500,000 bits per second. In comparison, sound files are transferred at a rate of 31,250 bits per second over MIDI.

RS-422 Interface Cable

If you are using the RS-422 interface cable, plug the *male* end into either of the Macintosh serial ports (**Printer** or **Modem**), and plug the *female* end into the computer interface port on the Emax rear panel. If you have a Macintosh Plus, you will need a short "DB9 to DIN 8" adapter to connect the cable to the Macintosh serial port. Select the serial port using the MIDI Setup dialog box (select **MIDI Setup** from the **Sampler** menu). Select **Emax RS-422** from the **Sampler** menu.

MIDI Interface

If you are using a Macintosh MIDI interface, connect the interface to either of the Macintosh serial ports (**Printer** or **Modem**), and connect **both** the MIDI IN and MIDI OUT cables to the Emax MIDI jacks. Select the serial port using the MIDI Setup dialog box (select **MIDI Setup** from the **Sampler** menu). Select **Emax MIDI** from the **Sampler** menu.

Sample Rate

The default Emax sample rate is 27777 kHz. When **Emax** is selected from the **Sampler** menu, Softsynth automatically enters 27777 in the *Sample Rate* parameter box each time you create a new parameter file. If you plan to play back the Softsynth sound on the Emax using a different sample rate, be sure to enter the correct value in the *Sample Rate* box.

Transferring sound files to the Emax

You must *synthesize* a Softsynth sound before you can transfer it to the Emax. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Once you have synthesized the sound, select **File->Sampler** from the **File** menu.

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the Emax, then click once on the **Open** button.

If the Macintosh is not able to communicate with the Emax for any reason, a "Can't Communicate" alert box will appear. If this happens, check the interface cable and serial port settings.

A dialog box will appear showing the name and length of the Softsynth sound file. The sound file can either *replace* an existing Emax voice, or be *added* to the selected preset.

Adding a New Sample

To add a new sample, click on the **Add New Sample** button, then use the vertical scroll bars next to "Preset:" and "Key:" to assign the new sample to a preset and a single key. The transferred sound can only be assigned to a

single key - use Sound Designer or the Emax's front panel to assign a wider keyboard range to the sample. Once you have selected a preset and key assignment, click once on the **OK** button.

Replacing an Existing Sample

To replace an existing Emax sample, click on the **Replace:** button. Next, select an Emax sample to replace using the vertical scroll bars next to "Preset:" and "Sample:". The name of each sample is displayed by *voice level* (PV = primary voice, SV = secondary voice) followed by *Sample Number* and *Sample Name*. Once you have selected a sample to replace, click on the **OK** button.

Sound Length

If the Macintosh sound file is longer than the selected Emax sample, a dialog box will appear asking you if you want to **Cancel** the transfer, **Truncate** the sound file or **Extend** the sound file. The extend option uses available "free" memory in the Emax to accomodate the transferred sound file. If the Emax does not have enough free memory to store the sound file, the end of the sound file will be truncated.

Roland S-10

The Roland S-10 is a twelve bit sampler with a maximum sampling time of 4 seconds (32 kHz sample rate) or 8 seconds (16 kHz sample rate). Softsynth transfers sounds to the S-10 at standard MIDI rate (31,500 bits per second).

Sample Rates

The S-10 has two sample rates: 31,250 Hz and 15,625 (they appear as 32 kHz and 16 kHz on the S-10 display). 31,250 is the most commonly used sample rate, so Softsynth automatically enters 31250 in the *Sample Rate* parameter box each time you create a new parameter file. Enter 15625 in the *Sample Rate* box if you plan to play back the sound at the S-10's lower sampling rate.

You must *synthesize* a Softsynth sound before you can transfer it to the S-10. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Transferring sound files to the S-10

Once you have synthesized the sound, select **File->Sampler** from the **File** menu.

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the S-10, then click once on the **Open** button.

A second dialog box will appear showing the name and length of the selected sound file, and a pair of up/down arrows for selecting an S-10 memory bank. The S-10 has four sound memory banks (**A**,**B**,**C** and **D**). Each bank has 32768 bytes of memory. Use the up/down arrows to select a destination bank (**A**,**B**,**C** or **D**) or string of banks (**AB**, **CD** or **ABCD**). After you select the destination bank(s), click on the **OK** button.

If the Macintosh is not able to communicate with the S-10 for any reason (e.g. the MIDI interface is not connected, the wrong serial port is selected, etc.), a "Can't Communicate" alert box will appear. If this happens, check the MIDI interface and serial port settings.

Sound Length

If the S-10 bank(s) does not have enough sound memory to store the Softsynth sound, a dialog box will appear to warn you that the Softsynth file will be truncated (cut-off). You have two options - you can **Cancel** the transfer or **Truncate** the sound file to fit the available memory space.

Akai S700/X7000

The Akai S700 and X7000 are twelve bit samplers with six separate banks of sample memory. Each bank has a maximum sampling time ranging from 1 second (32 kHz sample rate) to 8 seconds (4 kHz sample rate). Softsynth transfers sounds to the S700 and X7000 at the standard MIDI rate - 31,500 bits per second.

Sample Rates

Both samplers have variable sample rates. Softsynth's default sample rate is 31250. You can enter different values in the *Sample Rate* box, but these values must be between 4000 and 40,000 samples per second. The sampler playback sample rate is automatically set by Softsynth when a sound file is transferred.

You must *synthesize* a Softsynth sound before you can transfer it to the S700 or X7000. Selecting **Synthesize** from the **File** menu creates a sound file from the parameters you have entered, then saves it to disk.

Transferring Sound Files to the S700/X7000

Select **File->Sampler** from the **File** menu.

A dialog box will appear with a list of sound files available on the current disk drive. **Drive** and **Eject** buttons are provided for selecting the current drive or ejecting disks. Select the sound file you want to send to the sampler, then click once on the **Open** button.

A second dialog box will appear showing the name and length of the selected sound file, and a pair of up/down arrows for selecting an S700/X7000 memory bank. Both samplers have six sound memory banks. Each bank has 32768 bytes of memory. Use the up/down arrows to select a destination bank. After you select the bank(s) you want to transfer the sound file to, click on the **OK** button.

If the Macintosh is not able to communicate with the sampler for any reason (e.g. the MIDI interface is not connected, the wrong serial port is selected, etc.), a "Can't Communicate" alert box will appear. If this happens, check the MIDI interface and serial port settings.

If the sampler does not have enough free sound memory to store the Softsynth sound, a dialog box will appear to warn you that the Softsynth file will be truncated (cut-off). You have two options - you can **Cancel** the transfer or **Truncate** the sound file to fit the available memory space.

Optional Memory Expansion Board

Both the S700 and X7000 have an optional memory expansion board that increases the number of sample memory banks to sixteen. If you don't have the memory expansion board installed, **do not** attempt to transfer a sound file to banks 7 -16 (this will cause a communication error).

In addition to the additive synthesis capabilities described in the rest of this manual, Softsynth (version 2.0 or later) now includes a new FM Synthesis module. This module greatly expands Softsynth's flexibility and power, but it has not significantly changed the overall "look and feel" of the original Softsynth program.

Parameter File Compatibility

Parameter files created using version 1.0 of Softsynth are compatible with version 2.0, and vice-versa. Of course, parameter files created using version 1.0 will not have any FM "patching". You can use version 2.0 to add FM patching to a parameter file originally created using version 1.0. However, FM patching information stored in a parameter file will be ignored by Softsynth version 1.0.

What is FM Synthesis?

FM stands for *Frequency Modulation*. This synthesis technique has been used for many years, but has only recently gained widespread popularity due to the Yamaha DX™ and TX™ series FM synthesizers.

The principles of FM synthesis are generally more complex and difficult to understand than additive synthesis. Although this manual section will contain a brief description of FM synthesis, you should refer to one of the excellent books on FM synthesis (as implemented by the Yamaha DX/TX series). We recommend "The Complete DX7" by Howard Massie.

Additive vs. FM

Frequency modulation produces complex timbres (tone qualities) using comparatively few partials. However, when using FM you sacrifice the precise control over each frequency component you have when using purely additive synthesis. The combination of additive and FM techniques in the same synthesis system provides a very powerful tool for sound creation. But first we must understand FM synthesis!

In pure additive synthesis, oscillators (or *partials* in Softsynth's case) of different frequencies are mixed to produce a composite sound. FM synthesis adds another dimension to additive synthesis: partials can be mixed together (as in additive synthesis), but they can also *modulate* each other.

Modulators and Carriers

The basic "building blocks" of FM synthesis are *Modulators* and *Carriers*. A modulator is a waveform that modulates another waveform. A carrier is a waveform that is modulated by a modulator, but does not in turn modulate another waveform. Instead, it is mixed with other carriers to produce the overall sound.

In its simplest form, an FM patch consists of a single modulator that modulates a single carrier to produce the output sound. If we patch the carrier's output to the input of a second carrier, the first carrier becomes a modulator.

As we continue to add more modulators and carriers to our FM "patch", we can create a wider range of sounds. However, the signal path becomes more complex, and generally more confusing. It sometimes helps to draw a flow chart to keep track of your FM patching.

We have defined each modulator and carrier as a "waveform". In Softsynth's case, each waveform is actually a single partial, consisting of a frequency ratio, a waveshape, an amplitude envelope and a pitch envelope (see Section 3). Together, these parameters define the waveform's shape and initial frequency, as well as changes in amplitude and pitch as time passes.

Each of Softsynth's 32 partials can be used as a modulator, carrier or both. The result is an FM synthesizer with 32 partials, 40 stage envelopes, 15 stage pitch envelopes and 5 different wave shapes! Try to resist the temptation to use many partials and breakpoints in every sound - you can usually get the sound you want without enlisting all of Softsynth's considerable firepower!

Using Softsynth's FM Synthesis

Ok, let's learn how to create FM sounds using Softsynth. The first thing you may notice when you start the Softsynth program is the main screen - it looks exactly the same as version 1.0. So where is the FM? It is in a separate module. To display this module, select **FM Patching** from the pull-down **Edit** menu.

The FM Patching module has two horizontal rows of numbers ranging from 1 to 32. These numbers represent Softsynth's 32 partials. However, in this module they will be used as FM modulators and carriers.

The top row (labelled **Route modulator:**) allows you to select the partial(s) that will act as modulators. To select a partial for use as a modulator, click once on its number. When selected, the number will appear as "reverse" type inside a small black box.

The bottom row (labelled **To:**) allows you to select the partial you want to modulate. To select a partial to modulate, click once on its number. When selected, the number will appear as "reverse" type inside a small black box.

Creating a Modulation Patch

To create a modulation patch between two partials, one partial must be selected in each row. Once you have selected a modulator (top row) and a partial to modulate (bottom row), click once on the **Set** button. A line will appear connecting the two partials to indicate that the partial in the top row is modulating the partial in the bottom row.

Removing a Modulation Patch

To remove a modulation patch between two partials, select the pair of partials connected by the "patch" and click once on the **Clear** button. The connecting line should disappear to indicate that the patch has been removed.

"Stacking" Modulators

Once you have patched a modulator to modulate a partial, the modulated partial can be used as a carrier (included in the output sound), or it can act as a modulator and modulate another partial. A "stack" of modulators can be created by patching the modulated partial to modulate another partial. For example if partial 1 modulates partial 2, partial 2 can then modulate partial 3, 3 can modulate 4, etc.

This can be repeated until you have used up all 32 partials, creating an enormous stack. But don't try this now - it probably won't produce a very useful sound.

A single modulator can modulate many partials. To assign a modulator to modulate more than one partial, assign the first patch in the usual manner, then select another partial to modulate (leave the modulator the same) and click on the **Set** button. Patching lines will appear between the modulator and all of the partials it modulates.

However, a partial may not be *modulated* by more than one modulator. If you attempt to assign a second modulator to a partial that is already patched to a modulator, the original patch will be removed and the new patch will be assigned.

Including a Modulator in the Output Mix

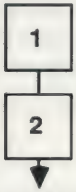
Occasionally, you may want to use a partial as a modulator **and** a carrier. Modulators are usually used as modulators only (you don't hear their sound in the output, just their effect on another partial), but Softsynth is able to include modulators in the sound's output mix. To do this, click on the check box labelled "Include modulator in output mix" after you have selected the modulation partials.

Feedback

Feedback is a commonly used FM synthesis technique in which the output of a modulator is connected to its input, or to the input of another modulator stacked "above" it. Softsynth allows you to patch a modulator's output to itself to create a single-stage feedback path, or around several stacked oscillators to create a multi-stage feedback path. Extreme amounts of feedback produce noise-like sounds, but smaller amounts can produce subtler changes in the sound's timbre. The feedback amount is controlled by the partial's modulation amount (see page S8).

Sample Patches

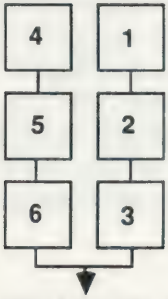
Block Diagram



Single modulator/carrier

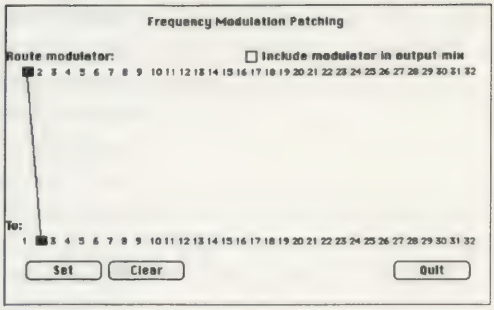
Stack of two modulators and a single carrier

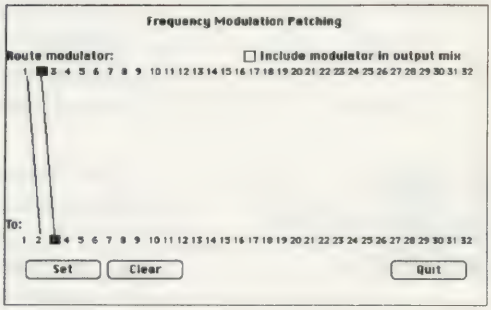
Two modulators, single carrier with feedback

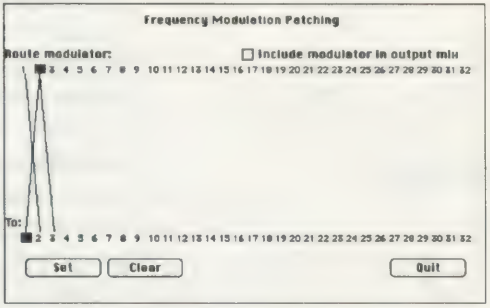


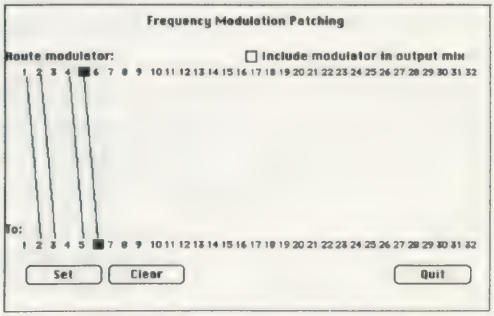
Parallel stacks, each with two modulators/one carrier

Equivalent Softsynth Patch









Algorithms

Many of you may already be familiar with the terminology used to describe an FM "patch" as implemented in the Yamaha DX/TX series synthesizers. The arrangement of modulators, carriers and feedback paths that we have referred to as a "patch", is called an *algorithm* by the folks at Yamaha. We prefer the term patch because Softsynth allows you to arrange operators in any configuration, while hardware-based FM synths usually provide a selection of "hard-wired" algorithms (the DX–7 has 32 algorithms).

These algorithms are represented on the DX–7's front panel as block diagrams showing signal flow. To help those of you familiar with the DX-7 understand Softsynth's FM patching, the illustration on page S9 shows some sample DX-7 algorithms and their equivalent Softsynth patches.

Once you have assigned all of the modulation patches for your sound, click once on the **Quit** button. This will return Softsynth to the main programming screen.

At this point, we have set up our basic FM patch. However, we must set two more parameters before synthesizing the sound: **Ratio** and **Modulation Amount**.

Ratio (frequency)

The **Ratio** parameter is described on page 27. It is the ratio of the individual partial's frequency to the sound's fundamental frequency. The ratio value for each partial can be edited in the single partial programming screen (see section 3).

If the partial is used as a modulator, its ratio determines the *modulation frequency*. This parameter is called "frequency" on the DX-7.

Modulation frequency plays an important role in determining the timbre of the modulated carrier in the output mix. Experiment with different modulation frequencies to understand their influence on timbre.

Modulation Amount

Modulation Amount (the extent to which a partial is modulated) can be independently adjusted for each modulator. The harmonic faders that appear in the main screen (see page 22) control the modulation amount of partials that act as modulators.

If a partial is not assigned an FM patch, its harmonic fader controls its level in the output sound. However, if a partial is used as a modulator, its harmonic fader controls the amount it modulates another partial.

If the modulator is also included in the output mix, the harmonic fader controls both the modulation amount and the partial's level in the output sound.

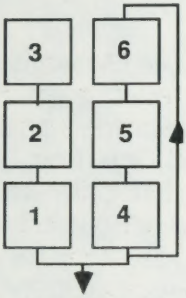
Higher fader levels cause *more* modulation. *Lower* fader levels cause *less* modulation.

Softsynth's modulation amount parameter has a somewhat different range than the DX–7's corresponding modulation index parameter. Softsynth's modulation amount has a range of 0 - 63, and the DX-7 has a modulation index range of 0 - 99. In both systems, the sensitivity of the modulation parameter varies throughout its range.

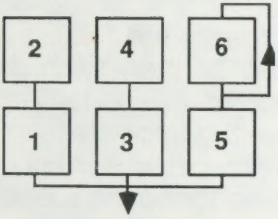
Start with relatively low modulation amount settings and increase them gradually. If you use excessive modulation, the resulting sound will contain lots of noise and high frequencies.

Sample DX-7 Patches

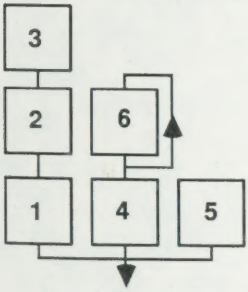
Block Diagram



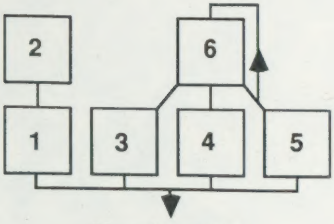
Algorithm #4



Algorithm #5

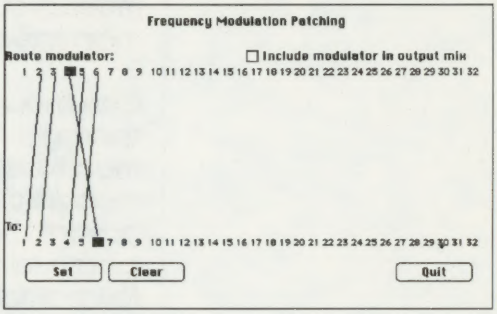


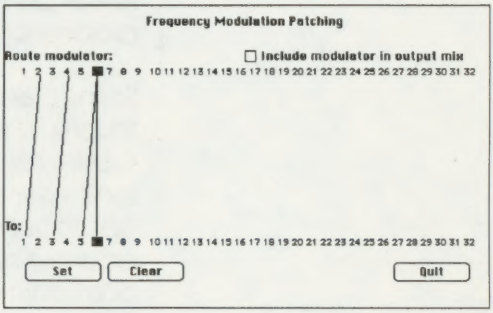
Algorithm #19

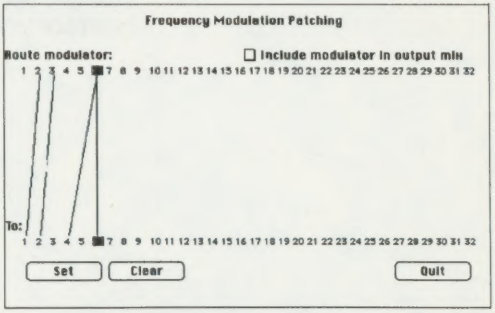


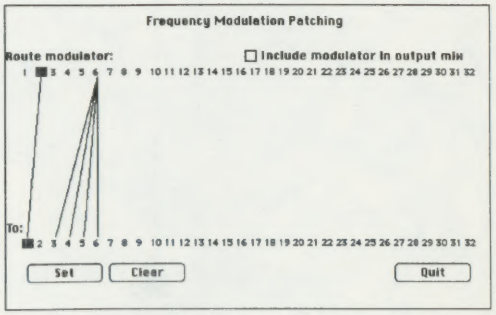
Algorithm #22

Equivalent Softsynth Patch









Which Partials are Modulators?

To help you remember which partials are modulators, a small box will appear around the number of each partial that is acting as a modulator (below the harmonic faders in the main screen).

One final word of advice - use FM synthesis sparingly. Start with only a few carriers and modulators, then add more carriers and modulators only as needed. The main advantage of FM synthesis is its ability to create fairly complex tones using relatively few partials. Take advantage of this by using fewer partials, and thus speeding up the process of synthesizing sound files.

Some people prefer to create the basic timbre of the sound using FM, then "finish" the subtler aspects of the sound using additive synthesis. Others choose the opposite approach - create the groundwork using additive synthesis, then enhance the sound's harmonic complexity with a little FM. Experiment, and find the approach that works best for you.